# ON THE AVOIDANCE OF RULE EXPLOSION IN FUZZY INFERENCE ENGINES

**Jeffrey J. WEINSCHENK[1], William E. COMBS[2], Robert J. MARKS II[3]**

[1] *Department of Electrical Engineering,*
*University of Washington, Seattle, WA, USA*
E-mail: weinschenk@ieee.org

[2]*The Boeing Company*
*Seattle, WA, USA*
E-mail: william.e.combs@boeing.com

[3]*Department of Electrical and Computer Engineering*
*Baylor University, Waco, TX, USA*
E-mail: r.marks@ieee.org

**Abstract**

Conventional fuzzy inference engines are constructed using an Intersection Rule Configuration (IRC) which suffers from *rule explosion.* Rule explosion, the phenomenon wherein the number of rules increases exponentially with the number of inputs, can often prohibit application of fuzzy inference engines to large Multi-Input, Multi-Output (MIMO) problems. Rule explosion can be mitigated through use of the more scalable Union Rule Configuration (URC). We derive a direct mapping between the IRC and single-layer URC for a class of additively separable problems. In addition, we prove that a two-layer URC is a universal approximator and can be applied to additively inseparable problems.

**Keywords**: fuzzy logic, rule explosion, complexity reduction, universal approximation, union rule configuration, intersection rule configuration

# 1 Introduction

Conventional fuzzy inference systems suffer from *rule explosion*, a phenomenon in which an increase in the number of antecedents results in the exponential growth of the number of fuzzy if-then rules. As a result, these inference systems do not scale well and it becomes impractical to construct large Multi-Input, Multi-Output (MIMO) systems. Complexity reduction techniques have been developed, however most of these techniques entail offline processing [1]. These techniques restrict fuzzy systems to static processes as rules cannot be adapted in real time.

In [2] Combs and Andrews suggest an alternative rule configuration that avoids rule explosion. Their method originates from the propositional logic expression

$$\left[(p \cap q) \Rightarrow r\right] \Leftrightarrow \left[(p \Rightarrow r) \cup (q \Rightarrow r)\right], \tag{1}$$

where $p$ and $q$ are antecedents, $r$ is a consequent, $\cap$ represents intersection, $\cup$ represents union, and $\Rightarrow$ is the implication operator. The left side of (1) is indicative of conventional fuzzy systems where multi-antecedent rules relate antecedent subsets to a consequent subset. Combs and Andrews refer to this rule structure as an Intersection Rule Configuration (IRC). Alternatively, the right side of (1) provides a novel rule architecture whereby the union of single antecedent rules yields a desired functionality. Combs and Andrews refer to this single antecedent rule structure as a Union Rule Configuration (URC). In [3], Yi *et al.* suggest a similar rule structure, composed of Single Input Rule Modules (SIRMs) which reduces to the URC under the constraint that each set of antecedent membership functions sums to unity.

As the URC was originally inspired by the propositional relation in (1), which is not an identity for fuzzy logic [4-8], it must be proved that the URC can successfully replace multi-antecedent rules with single antecedent ones while still retaining an expert's perceived correlation between the antecedents. Indeed, a similar rule architecture given in [9] was criticized for its inability to act as a universal approximator [10-11]. This paper proves that the single-layer form of the URC structure suggested by Combs and Andrews does not support universal approximation. However, a direct mapping does exist between the IRC and the single-layer URC for a special class of problems [12]. More complex problems can be addressed through application of a multi-layer URC, which is a universal approximator [13].

The IRC and single-layer URC architectures are briefly reviewed in Section II. In Section III we derive a mapping between the IRC and the

single-layer URC architectures that holds for a certain class of problems. In Section IV, we investigate a two-layer URC architecture that is applicable to a larger class of problems. A simple regression example is given in Section V which contrasts the IRC and URC. Concluding remarks follow in Section VI.

## 2 A Review of the IRC and Single-Layer URC Architectures

Consider a Multiple Input, Single Output (MISO) IRC fuzzy system in which the t-norm is implemented as a scalar product, the t-conorm is implemented as a bounded sum, and the implication operator is implemented as a scalar product (Larsen implication). For $P$ antecedents and centroid defuzzification, the MISO IRC output formula is expressed as,

$$y_{IRC}(\mathbf{x}) = \frac{\sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} T(a_1, \cdots, a_P) \prod_{j=1}^{P} \mu_{j,a_j}(x_j)}{\sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} \prod_{j=1}^{P} \mu_{j,a_j}(x_j)}, \tag{2}$$

where $\mu_{j,a}(x_j)$ is an element of a vector of membership values (one value per antecedent subset) for the $i^{\text{th}}$ antecedent and each antecedent contains $N_i$ subsets, each indexed by $a_i$. The matrix $\mathbf{T}$ denotes a $P$-dimensional rule matrix and a single element, denoted as $T(a_1,\ldots,a_P)$, indicates the desired consequent subset via a scalar-valued center of mass. Initially, the min function in the bounded sum t-conorm is neglected (a constraint will be derived that guarantees the sum is always less than or equal to one). Using the equality

$$\sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} \prod_{j=1}^{P} \mu_{j,a_j}(x_j) = \prod_{j=1}^{P} \sum_{i=1}^{N_j} \mu_{j,i}(x_j), \tag{3}$$

the denominator in (2) can be factored into the numerator such that

$$y_{IRC}(\mathbf{x}) = \sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} T(a_1, \cdots, a_P) \prod_{j=1}^{P} \hat{\mu}_{j,a_j}(x_j), \tag{4}$$

where $\hat{\mu}_{i,a_i}$ is a sum-normalized version of $\mu_{i,a_i}$ or

$$\hat{\mu}_{j,a_j}(x_j) = \frac{\mu_{j,a_j}(x_j)}{\sum_{i=1}^{N_j} \mu_{j,i}(x_j)}. \tag{5}$$

The IRC inference engine is a universal approximator [14-16].

In contrast, the output formula for a single-layer MISO URC fuzzy system with centroid defuzzification, using the same t-norm, t-conorm, and implication operators is given by

$$y_{URC}(\mathbf{x}) = \frac{\sum_{i=1}^{P} \sum_{j=1}^{N_i} z_{i,j} \mu_{i,j}(x_i)}{\sum_{i=1}^{P} \sum_{j=1}^{N_i} \mu_{i,j}(x_i)}, \tag{6}$$

where each antecedent subset is associated with a consequent subset whose center of mass is $z_{i,j}$.[1] Although this output formula is not given in [2], it is readily attainable from the right side of (1). Fig. 1 compares the IRC and single-layer URC rule tables for a two antecedent inference system. The linguistic labels for the antecedent and consequent subsets are given in Fig. 2. In essence, the single-layer URC fuzzy inference engine differs from the IRC inference engine in that it only utilizes single antecedent rules. Thus each antecedent subset is mapped to a single consequent subset, independent of all other antecedents.

Finally, observe that the MISO IRC contains one multi-antecedent rule per element in **T**, or

$$\text{Number of IRC rules} = \prod_{i=1}^{P} N_i \,. \tag{7}$$

Similarly, the single-layer MISO URC contains one single-antecedent rule

[1] Chen *et al.* provide a similar output formula [9].

per membership vector element, such that

$$\text{Number of URC rules} = \sum_{i=1}^{P} N_i \,, \tag{8}$$

as there are $P$ membership vectors of $N_i$ subsets each. Indeed, with respect to an increase in the number of antecedents the single-layer URC achieves a linear growth in the number of fuzzy rules while the IRC suffers an exponential growth in the number of fuzzy rules.

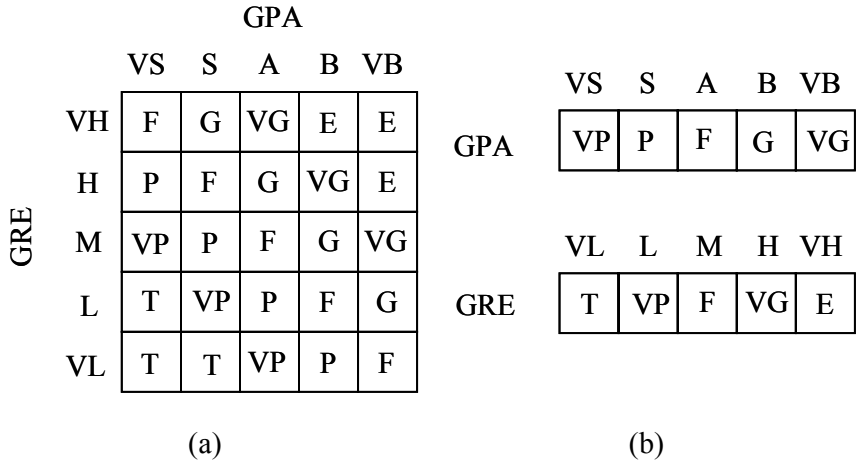


**Figure 1.** (a) An IRC rule table that gives a measure of the strength of a graduate school applicant based on GRE and GPA scores. (b) Single-layer URC rule tables that rate the applicant's strength based on GRE and GPA scores. Notice that each antecedent subset is mapped to a single consequent subset, independent of all other antecedents.

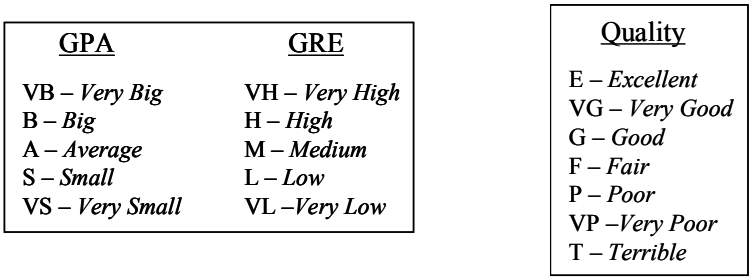


**Figure 2.** The linguistic identifiers for the antecedent and consequent subsets.

## 3 A Mapping for Additively Separable IRC Rule Tables

In this section, a mapping between the IRC and Single-Layer URC (SLURC) is derived. We begin with the single-layer URC output formula in (6) and derive necessary constraints for equality between the IRC and single-layer URC. First, note that

$$\sum_{i=1}^{N_j} \hat{\mu}_{j,i}(x_j) = 1, \tag{9}$$

which follows directly from (5) and more generally,

$$\sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} \prod_{j=1}^{P} \hat{\mu}_{j,a_j}(x_j) = \prod_{j=1}^{P} \sum_{i=1}^{N_j} \hat{\mu}_{j,i}(x_j) = 1. \tag{10}$$

Each term in the numerator of (6) is multiplied by unity in a form similar to the left-most side of (10) such that,

$$y_{SLURC}(\mathbf{x}) = \frac{\displaystyle\sum_{i=1}^{P} \sum_{a_i=1}^{N_i} z_{i,a_i} \mu_{i,a_i}(x_i) \left( \sum_{a_1=1}^{N_1} \cdots \sum_{a_{i-1}=1}^{N_{i-1}} \sum_{a_{i+1}=1}^{N_{i+1}} \cdots \sum_{a_P=1}^{N_P} \prod_{\substack{k=1 \\ k \neq i}}^{P} \hat{\mu}_{k,a_i}(x_k) \right)}{\displaystyle\sum_{i=1}^{P} \sum_{j=1}^{N_j} \mu_{i,j}(x_i)}, \tag{11}$$

where the subscript $j$ is replaced with the subscript $a_i$ in the numerator. Moving the factor $z_{i,a}\mu_{i,a}(x_i)$ inside the sums results in

$$y_{SLURC}(\mathbf{x}) = \frac{\displaystyle\sum_{i=1}^{P} \sum_{a_1=1}^{N_1} \cdots \sum_{a_P=1}^{N_P} z_{i,a_i} \mu_{i,a_i}(x_i) \prod_{\substack{k=1 \\ k \neq i}}^{P} \hat{\mu}_{k,a_i}(x_k)}{\displaystyle\sum_{i=1}^{P} \sum_{j=1}^{N_i} \mu_{i,j}(x_i)}. \tag{12}$$

Next, $\mu_{i,a_i}(x_i)$ is substituted according to (5) to yield

$$y_{SLURC}(\mathbf{x}) = \frac{\sum_{i=1}^{P}\sum_{a_1=1}^{N_1}\cdots\sum_{a_P=1}^{N_P} z_{i,a_i}\left(\sum_{l=1}^{N_i}\mu_{i,l}(x_i)\right)\prod_{k=1}^{P}\hat{\mu}_{k,a_i}(x_k)}{\sum_{i=1}^{P}\sum_{j=1}^{N_i}\mu_{i,j}(x_i)}. \quad (13)$$

Rearranging the sums and distributing the denominator into the numerator gives

$$y_{SLURC}(\mathbf{x}) = \sum_{a_1=1}^{N_1}\cdots\sum_{a_P=1}^{N_P}\sum_{i=1}^{P}\left[\frac{z_{i,a_i}\left(\sum_{l=1}^{N_i}\mu_{i,l}(x_i)\right)}{\left(\sum_{i=1}^{P}\sum_{j=1}^{N_i}\mu_{i,j}(x_i)\right)}\right]\prod_{k=1}^{P}\hat{\mu}_{k,a_i}(x_k). \quad (14)$$

Compare (14) to the output formula for the IRC in (4) and it is apparent that for equality to exist between the IRC and single-layer URC, the IRC rule table **T** must be equal to the quantity contained within the square brackets in (14). This constraint can also be expressed in terms of a single element of the IRC rule table as

$$\mathbf{T}(a_1,\cdots,a_P) = \sum_{i=1}^{P}\frac{z_{i,a_i}\left(\sum_{l=1}^{N_i}\mu_{i,l}(x_i)\right)}{\left(\sum_{n=1}^{P}\sum_{n=1}^{N_{mi}}\mu_{n,m}(x_n)\right)}. \quad (15)$$

Although this constraint guarantees equality, it is somewhat problematic as the consequent centers of mass (elements of **T**) are dependent on the inputs. This dependency is removed when the membership functions are sum-normalized such that

$$\sum_{j=1}^{N_i}\mu_{i,j}(x_i) = 1 \quad \forall\ i. \quad (16)$$

In addition, this constraint also guarantees the sum of all combinations of membership values for the *P* antecedents is unity

$$\sum_{a_1=1}^{N_1}\cdots\sum_{a_P=1}^{N_P}\prod_{j=1}^{P}\mu_{j,a_j}(x_j)=\prod_{i=1}^{P}\sum_{j=1}^{N_i}\mu_{i,j}(x_i)=1\ . \tag{17}$$

Thus the sum of any subset of these combinations can yield a value of at most unity. Therefore, the min function in the bounded sum t-conorm can be discarded when writing the output formula for the IRC in (2). Given the constraint in (16), (15) can be rewritten as

$$T(a_1,\cdots,a_P)=\sum_{i=1}^{P}\frac{z_{i,a_i}}{P}\ . \tag{18}$$

Defining

$$v_{i,j}=\frac{z_{i,j}}{P}\quad\forall\ i,j\ , \tag{19}$$

we refer to the set of $v_{i,j}$ as *projection vectors* in which case (18) becomes

$$T(a_1,\cdots,a_P)=\sum_{i=1}^{P}v_{i,a_i}\ . \tag{20}$$

Any IRC rule table that satisfies (20) is said to be *additively separable*. Note the set of projection vectors $[v_1,v_2,\ldots,v_P]$ is not unique however as $[v_1+c_1,v_2+c_2,\ldots,v_P+c_P]$ is an equivalent set provided that the constants $c_1$, $c_2,\ldots,\ c_P$ sum to zero. Thus the single-layer URC inference engine can *only* implement those IRC rule tables that are additively separable. As IRC inference systems with additively separable rule tables can only construct additively separable output surfaces, single-layer URC systems can only generate additively separable output surfaces (i.e. surfaces expressible as a sum of univariate functions). Due to this limitation the *single-layer URC inference engine is not a universal approximator*. The output formula for the single-layer URC is written in terms of projection vectors using (19) to yield

$$y_{SLURC}(\mathbf{x})=\sum_{i=1}^{P}\sum_{j=1}^{N_i}v_{i,j}\hat{\mu}_{i,j}(x_i)\ . \tag{21}$$

Recall the number of fuzzy rules grows exponentially in IRC fuzzy systems as shown in (7) while the number of rules grows linearly in single-layer URC

fuzzy systems as shown in (8). Thus, rule explosion is completely eliminated for additively separable problems through use of a single-layer URC architecture.

Additive separability is an important simplifying property for fuzzy inference systems that requires each antecedent have a direct relationship with the consequent, independent of the other antecedents. As an example, two additively separable IRC rule tables are depicted with their projection vectors in Fig. 3. Notice that the projection vectors can be computed in a variety of ways including averaging over the rows and columns of the rule table and subtracting off $1/P^{\text{th}}$ of the mean of the entire rule table from each vector. Alternatively, one may select the projection vectors to be the first row and column of the rule table minus an offset corresponding to $1/P^{\text{th}}$ the element $T(1,1,\ldots,1)$. One can test for additive separability by forming projection vectors as just described and then reconstructing the IRC rule table via an outer sum of the projection vectors—the IRC rule table is additively separable if it matches the reconstructed rule table.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 0 |
| 1 | 2 | 3 | 4 | 5 | 1 |
| 2 | 3 | 4 | 5 | 6 | 2 |
| 3 | 4 | 5 | 6 | 7 | 3 |
| 4 | 5 | 6 | 7 | 8 | 4 |
| 0 | 1 | 2 | 3 | 4 | |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 0 |
| 1 | 2 | 3 | 4 | 5 | 1 |
| 2 | 3 | 4 | 5 | 6 | 2 |
| 1 | 2 | 3 | 4 | 5 | 1 |
| 0 | 1 | 2 | 3 | 4 | 0 |
| 0 | 1 | 2 | 3 | 4 | |

**Figure 3.** Two additively separable rule tables where each square is occupied by a consequent center of mass. The projection vectors are given along the vertical and horizontal axes. Notice that each matrix element is the sum of the corresponding vertical and horizontal components.

## 4 Additively Inseparable Problems and the Multi-Layer URC

As the single-layer URC can only implement additively separable output surfaces, another structure is required that alleviates rule explosion for more complex problems. To motivate the derivation of one such structure, we examine a similar problem from another field. In 1969 Minsky and Pappert demonstrated that a feed forward neural network with no hidden layers is not a universal approximator; instead, it can only correctly classify linearly separable data sets [17]. However with the addition of a least one hidden layer (for a total of two or more layers) the feed-forward neural network becomes a universal approximator [18]. For our purposes it is convenient to examine a nonstandard form of the output formula for a single neuron

$$y_{NN}(\mathbf{x}) = \sum_{i=1}^{P} w_i \, q_i(x_i), \tag{22}$$

where $P$ represents the number of inputs to the node, the $w_i$'s are weights, and the $q_i(x_i)$'s are nonlinear kernel functions. This formula can be derived from a multi-layer feed-forward neural network by treating the output kernel functions of one layer as input kernel functions of the next.

As the single-layer feed-forward neural network becomes a universal approximator with the addition of another hidden layer, we wondered whether the same effect could be realized by cascading two URC layers. Graphically, the two-layer neural network and a two-layer URC are quite similar (see Fig. 4 and Fig. 5). The most readily noticeable difference between the layered URC fuzzy system and the multi-layer neural network is that the layered URC fuzzy system applies a *set* of membership functions to each input, whereas the feed-forward neural network applies a *single* kernel function to each input. Interestingly, the output formula of the single-layer URC simplifies to

$$y_{SLURC}(\mathbf{x}) = \sum_{i=1}^{P} v_{i,1} \, \mu_{i,1}(x_i), \tag{23}$$

when $v_{i,j} = 0$ for $j > 1$. Notice we write $\mu_{i,1}$ as the remaining $\mu_{i,j}$ for $j > 1$ can be chosen such that the set is sum-normal. In comparing (22) and (23) we see the MISO URC subsumes the neuron. In addition, the URC also accommodates a neural network bias term (not shown in (22)) as

$$y_{SLURC}(\mathbf{x}) = \sum_{i=1}^{P} v_{i,1}\, \mu_i(x_i) + v_{P+1,1}\,, \tag{24}$$

where the $(P+1)^{st}$ input has only one membership function, $\mu_{P+1,1}(x_{P+1}) = 1$, and the single element $v_{P+1,1}$ is the bias.

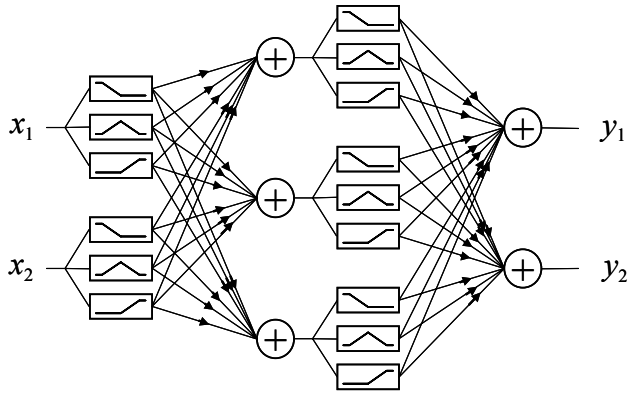


**Figure 4.** A two-layer URC fuzzy system with three subsets per antecedent.

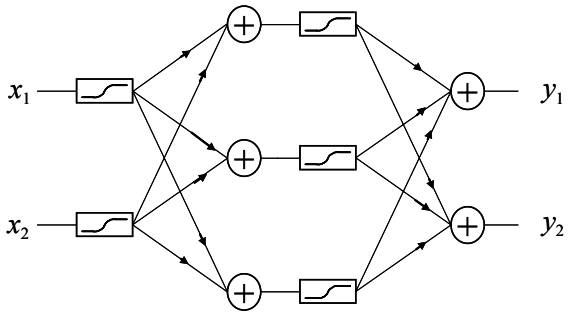


**Figure 5.** A two-input, two-output, two-layer feed-forward neural network.

As a result, the neuron can be viewed as a degenerate form of the more general single-layer MISO URC. Most kernel functions used in neurons (e.g. sigmoids, radial basis functions, etc.) can likewise be viewed as membership functions of a MISO URC. Thus a single-layer MIMO URC can implement a neural network layer and a cascade of MIMO URC layers can implement an entire feed-forward neural network. Therefore as the feed-forward neural network containing at least one hidden layer and constructed with either sigmoidal or non-sigmoidal kernel functions is a universal approximator [18-19], a cascade of two or more MIMO URC layers is also a universal

approximator because it may implement any neural network of corresponding size.

The fact that a two-layer MIMO URC is a universal approximator means that it is capable of producing any output surface generated including those which are additively inseparable. Given the efficiency of the single-layer URC, we wondered if the two-layer URC could implement additively inseparable output surfaces using fewer rules than the analogous IRC inference system. As a single MIMO URC layer can only construct additively separable output surfaces the additively inseparable problem must be decomposed into two additively separable problems—one problem per URC layer. That is the first URC layer must convert its inputs (via an additively separable transformation) to a set of intermediate variables that relate to the desired output through a second additively separable transformation.

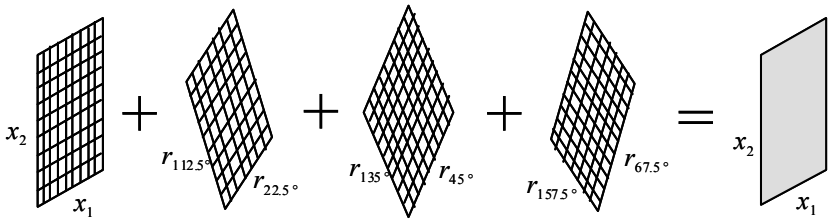


**Figure 6.** Demonstration of tomographic decomposition. Each rotated surface is additively separable; the superposition of the rotated surfaces yields the desired output surface.

One strategy for constructing a two-layer URC is based on tomography. In the field of tomography it is well-known that a surface can be reconstructed to an arbitrary accuracy from a set of one dimensional projections of that surface through many different angles [20]. For example, a two dimensional additively inseparable surface can be reconstructed by forming two dimensional additively separable surfaces (each oriented at a different angle from the others) and then superimposing these two dimensional surfaces (Fig. 6). Indeed Ashrafzadeh *et. al*. have observed that some simple, symmetric additively inseparable surfaces can become additively separable when rotated by 45 degrees [21]. Tomographic reconstruction is accomplished by

$$f(\mathbf{x}) \approx c\sum_{k=1}^{K} \hat{f}(\chi(\mathbf{x}, \boldsymbol{\theta}_k)) \ , \tag{25}$$

Where $f(\mathbf{x})$ is the desired multidimensional surface, the tilde form is a one dimensional filtered back-projection, and $\chi(\mathbf{x},\boldsymbol{\theta}_k)$ is an axis running through the space oriented according to angle vector $\boldsymbol{\theta}_k$ [22]. The quantity $\chi(\mathbf{x},\boldsymbol{\theta}_k)$ is constructed by

$$\chi(\mathbf{x},\boldsymbol{\theta}_k) = \mathbf{x}\cdot\boldsymbol{\varepsilon}(\boldsymbol{\theta}_k) \tag{26}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$ is a unit vector that indicates the orientation of the new axis, an element of which is given by

$$\varepsilon_i(\boldsymbol{\theta}_k) = \begin{cases} \left(\cos\theta_{k,1}\right)\left(\sin\theta_{k,2}\right)\cdots\left(\sin\theta_{k,P-1}\right) & i=1 \\ \left(\sin\theta_{k,1}\right)\cdots\left(\sin\theta_{k,P-1}\right) & i=2 \\ \left(\cos\theta_{k,i-1}\right)\left(\sin\theta_{k,i}\right)\cdots\left(\sin\theta_{k,P-1}\right) & 2<i<P \\ \left(\cos\theta_{k,i-1}\right) & i=P \end{cases}. \tag{27}$$

with $\boldsymbol{\theta}_{k,i}$ representing the $i$th element of the vector $\boldsymbol{\theta}_k$. A filtered back-projection is obtained using

$$\hat{f}(\chi) = \int_{-\infty}^{\infty} \alpha_{\boldsymbol{\theta}_k} \,|\rho|^{P-1}\, F(\rho,\boldsymbol{\theta}_k) e^{j2\pi\rho\chi}\, d\rho \tag{28}$$

where $F(\rho,\boldsymbol{\theta}_k)$ is the multidimensional Fourier transform of $f(\mathbf{x})$ in polar coordinates and the alpha term comes from the Jacobian of the transformation

$$\alpha_{\boldsymbol{\theta}_k} = \left|\sin(\theta_{k,2})\sin^2(\theta_{k,3})\sin^3(\theta_{k,4})\cdots\sin^{P-2}(\theta_{k,P-1})\right|. \tag{29}$$

A two-layer URC inference engine can implement (25) if the first layer generates the rotated variables $\chi(\mathbf{x},\boldsymbol{\theta}_k)$ at the discrete angles $\boldsymbol{\theta}_k$ and the second layer uses the rotated variables to form and superimpose one dimensional filtered projections. The first URC layer constructs rotated variables according to

$$\chi(\mathbf{x},\boldsymbol{\theta}_k) = \sum_{i=1}^{P} \varepsilon_i(\boldsymbol{\theta}_k) x_i = \sum_{i=1}^{P}\sum_{j=1}^{N_i} v_{i,j}\hat{\mu}_{i,j}(x_i) \tag{30}$$

where the membership functions can be thought of as interpolation kernels and the $v_{i,j}$ are interpolation weights. Notice this operation is simply a weighted sum performed on normalized inputs and need not be performed in a URC layer at all; rather, it could be viewed as a pre-processing step as in [23]. The second URC layer constructs and superimposes the one dimensional filtered projections according to

$$y(\chi(\mathbf{x},\boldsymbol{\theta}_k)) = \sum_{k=1}^{P}\sum_{j=1}^{M_k} w_{k,j}\hat{\mu}_{k,j}(\chi(\mathbf{x},\boldsymbol{\theta}_k)). \tag{31}$$

In general, the optimum rotation angles, $\boldsymbol{\theta}_k$, can be determined by standard search techniques [24]. Additional methods are explored in [22, 25].

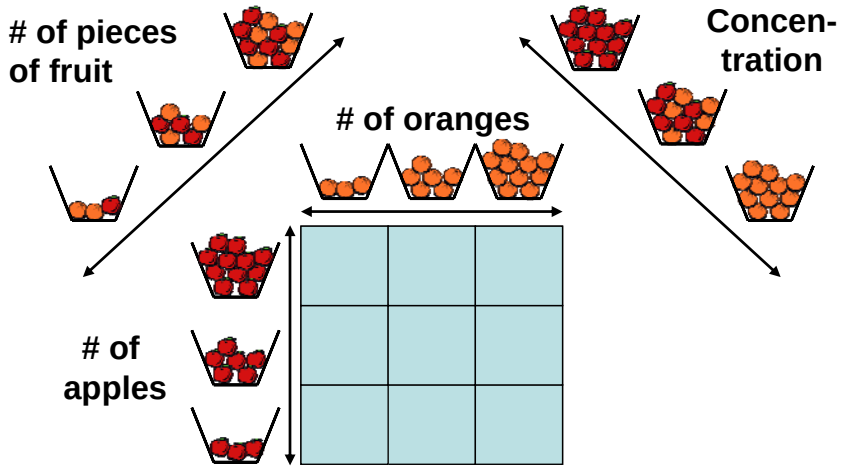


**Figure 7.** Mixing apples and oranges: fuzzy linguistic meaning is preserved in rotated variables.

Despite the rule table rotations, some linguistic meaning is retained in the two-layer URC inference system. Consider the pedagogical example shown in Fig. 7, where the first input gives a fuzzy measure of the number of apples in a basket and the second input gives a fuzzy measure of the number of oranges in the basket. In the case of a $\pi/4$ radian rotation of the inputs, a new variable is constructed that gives a fuzzy measure of the number of pieces of fruit in the basket. Similarly, a rotation of $-\pi/4$ radians gives a fuzzy measure of the concentration of oranges (relative to apples) in the basket. In general, rotations of inputs amount to the mixing of apples and oranges where

rotations through angles other than $\pm\pi/4$ radians yield weighted fuzzy measures of sum and difference.

The precise amount of computational savings provided by the layered URC is dependent upon the complexity of the desired output surface; however the savings is exponential for output surfaces that are accurately represented by few projections. In the worst case, the two-layer URC fuzzy system may require more resources than the analogous IRC inference system. In the next section we explore a few simple problems to contrast the IRC and URC. In addition, we demonstrate how additional complexity reduction can be achieved by using more than two URC layers. Development of a generalized, *N*-layer URC inference system is left for future research.

# 5 Implementation Examples

In this section the IRC and URC inference engines are contrasted through two pedagogical examples. The first example highlights the complexity savings offered by a single-layer URC for an additively separable problem. The second example demonstrates the complexity reduction potential of the two-layer URC inference system for additively inseparable problems.

## 5.1 The Single-Layer URC

Consider an IRC fuzzy system with two antecedents and a single consequent whose purpose is to rate applicants to a graduate program (Fig. 8). Let the first antecedent represent the quality of an applicant's Graduate Record Examination (GRE) with subsets of 'High,' 'Medium,' and 'Low.' Let the second antecedent be the quality of an applicant's undergraduate Grade Point Average (GPA). For the sake of clarity, different names are given to the subsets of GPA: 'Above Average,' 'Average,' and 'Below Average.' The consequent represents the quality of an applicant with subsets of 'Excellent,' 'Very Good' 'Good,' 'Fair,' 'Poor,' and 'Very Poor'. As the only distinguishing characteristic of a consequent subset is its center of mass, we use singleton consequent subsets and write the center of mass for each directly in the rule table. The antecedent membership functions are assumed to be sum-normalized, but are not shown as they have no direct bearing on the comparison.

Notice that the IRC rule table of Fig. 8 is additively separable as defined by (20). Therefore projection vectors can be selected that exactly reconstruct the IRC rule table. The projection vectors are selected to be the first row and column of the rule table minus an offset corresponding to 1/2 the element

$T(1,1)$ (Fig. 9). Notice some of the original consequent subsets are no longer required, while one new consequent subset is added (i.e. 'Reject').

| | | **GPA** | | |
|---|---|---|---|---|
| | | AA | A | BA |
| **GRE** | H | E\|10 | VG\|8 | G\|6 |
| | M | G\|6 | F\|4 | P\|2 |
| | L | F\|4 | P\|2 | VP\|0 |

**Figure 8.** An additively separable IRC rule table. The consequent centers of mass are listed next to each consequent subset in the rule table.

| | | | |
|---|---|---|---|
| **GRE** | H→E\|10 | M→P\|2 | L→R\|-2 |
| **GPA** | AB→E\|10 | A→G\|6 | BA→P\|2 |

**Figure 9.** Single-layer URC rule table. The symbol → represents Larsen implication. Singleton consequent membership sets are used; the centers of mass are recorded in the rule table next to each consequent set name.

| | | Subsets per Antecedent | | | |
|---|---|---|---|---|---|
| | | 2 | 5 | 10 | 50 |
| # of Inputs | 2 | 4 (**4**) | 25 (**10**) | $10^2$ (**20**) | $2.5 \cdot 10^3$ (**100**) |
| | 5 | 32 (**10**) | $3.13 \cdot 10^3$ (**25**) | $10^5$ (**50**) | $3.13 \cdot 10^8$ (**250**) |
| | 10 | $1.02 \cdot 10^3$ (**20**) | $9.77 \cdot 10^6$ (**50**) | $10^{10}$ (**100**) | $9.77 \cdot 10^{16}$ (**500**) |
| | 50 | $1.26 \cdot 10^{15}$ (**100**) | $8.88 \cdot 10^{34}$ (**250**) | $10^{50}$ (**500**) | $8.88 \cdot 10^{84}$ (**$2.5 \cdot 10^3$**) |
| | 100 | $1.27 \cdot 10^{30}$ (**200**) | $7.89 \cdot 10^{69}$ (**500**) | $10^{100}$ (**$10^3$**) | $7.89 \cdot 10^{169}$ (**$5 \cdot 10^3$**) |

**Figure 10.** Comparison of the number of rules contained in the IRC and single-layer URC inference systems (for additively separable problems). The numbers of rules required by the single-layer URC system are shown in bold.

While the IRC and single-layer URC inference systems implement the same output surface, the complexity of the single-layer URC inference system is significantly less than that of the IRC inference system. In Fig. 10 the number of rules required by the IRC and single-layer URC fuzzy systems, given by (7) and (8) respectively, are compared for various numbers of antecedents and subsets per antecedent. Notice that the IRC inference system quickly becomes

system quickly becomes intractable while the single-layer URC inference system remains functional. Even for a moderately sized MIMO system containing 10 antecedents with 5 subsets each, the single-layer URC requires 99.9995% fewer rules than the analogous IRC system!

## 5.2 The Two-Layer URC

In this subsection we explore the fuzzy XOR to contrast the IRC and two-layer URC inference systems. Consider the function

$$g(x_1, x_2) = 0.5 - 2(x_1 - 0.5)(x_2 - 0.5) \tag{32}$$

which implements a fuzzy XOR. For visualization purposes, a fuzzy XOR can be thought of as a system that controls the luminosity of a dual-switch dimmer light. Notice $g(x_1,x_2)$ satisfies the boundary conditions $g(0,0) = g(1,1) = 0$ (i.e. the light is 'off' if the switches are either both 'on' or both 'off') and $g(0,1) = g(1,0) = 1$ (the light is 'on' if a single switch is 'on'). An IRC fuzzy system is constructed that approximates (32) by forming the IRC rule table from samples of (32). In this case, (32) is essentially interpolated from its samples by the IRC inference system and the shape of the antecedent membership functions determines the type of interpolation. The rule table for one such IRC inference system is shown in Fig. 11. Triangular membership functions are used (Fig. 12).

| $x_2$ \ $x_1$ | LOW | MED | HIGH |
|---|---|---|---|
| HIGH | 1 | 0.5 | 0 |
| MED | 0.5 | 0.5 | 0.5 |
| LOW | 0 | 0.5 | 1 |

**Figure 11.** An additively inseparable IRC rule table that implements the fuzzy XOR where the consequent sets are referred to by their centers of mass.

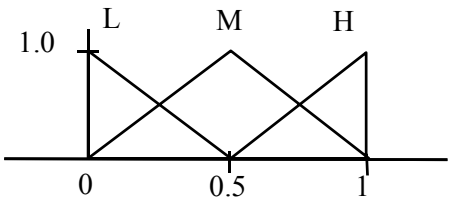

**Figure 12.** Triangular antecedent membership functions give rise to linear interpolation.

The first step in constructing a two-layer URC fuzzy system is to determine which projections are best suited for the reconstruction of the desired output surface. Standard search techniques are one option; however for this simple problem observation will suffice. Rotation of the desired output surface (given in (34)) by $\pi/4$ radians about the point (0.5, 0.5) yields

$$g_{\pi/4}(x_1, x_2) = 0.5 + 2\left(x_1 - 0.5\right)^2 - 2\left(x_2 - 0.5\right)^2 . \tag{33}$$

Notice this surface is additively separable and may be perfectly reconstructed by summing two one-dimensional functions. Therefore a two-layer URC fuzzy system can approximate (32) if the first layer generates $\pm\pi/4$ radian rotations of the inputs and the second layer implements the additively separable surface given in (33). In general, more complex surfaces may require additional pairs of orthogonal projections.

| | $x_1$ | | $X_2$ | |
|---|---|---|---|---|
| | LOW | HIGH | LOW | HIGH |
| $r_{\pi/4}$ | 0 | 0.5 | 0 | 0.5 |
| $r_{-\pi/4}$ | 0 | 0.5 | 0.5 | 0 |

**Figure 13.** Consequent centers of mass for the first layer of the two-layer URC inference system.

The first URC layer is constructed such that each antecedent has two subsets, 'Low' and 'High'. In addition, the first layer has two consequents: the first consequent, $r_{\pi/4}$, represents a rotated input oriented at an angle of $\pi/4$ radians with respect the original $x_1$ axis and the second consequent, $r_{-\pi/4}$, represents another rotated input oriented at an angle of $-\pi/4$ radians with respect to the $x_1$ axis. The consequent centers of mass are computed according to (26). The resulting URC rule table is provided in Fig. 13 and the antecedent membership functions are shown in Fig. 14. The second layer rule table is obtained by sampling the univariate functions that compose (33). Fig. 15 gives the consequent centers of mass for the second layer URC inference system and Fig. 16 shows the antecedent membership functions.

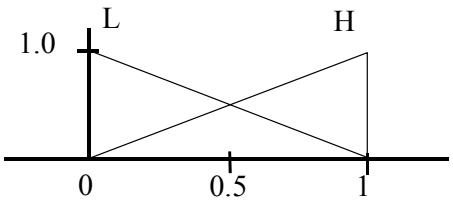


**Figure 14.** The antecedent membership functions for the first layer of the two-layer URC inference system.

| $r_{\pi/4}$ | | | $r_{-\pi/4}$ | | |
|---|---|---|---|---|---|
| LOW | MED | HIGH | LOW | MED | HIGH |
| 0 | 0.5 | 0 | 0.5 | 0 | 0.5 |

**Figure 15.** Consequent centers of mass for the second layer of the two-layer URC inference system.

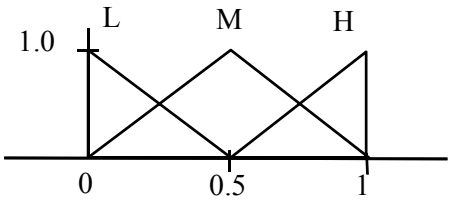


**Figure 16.** The antecedent membership functions for the second layer of the two-layer URC inference system.

The output surface of the IRC inference system is shown in Fig. 17 (which is also identical to (32)). Fig. 18 depicts the output surface of the two-layer URC fuzzy system for the case where the antecedents in the second layer possess three subsets each. Notice that the output surfaces shown in Figures 17 and 18 have the same general shape, yet are not identical. Both output surfaces are the result of linear interpolation because triangular membership functions are used for the antecedents of both systems. However, the IRC fuzzy system forms an output surface by directly linearly interpolating between samples from the original surface given in (32). Conversely, the two-layer URC fuzzy system effectively performs linear interpolation on samples from the rotated surface given in (33) and subsequently rotates the result back to the proper orientation. Indeed both are valid approximations and had the desired surface been the one shown in Fig. 18 the IRC and two-layer URC fuzzy systems would still yield the same output—yet it would be the IRC fuzzy system that is less accurate.

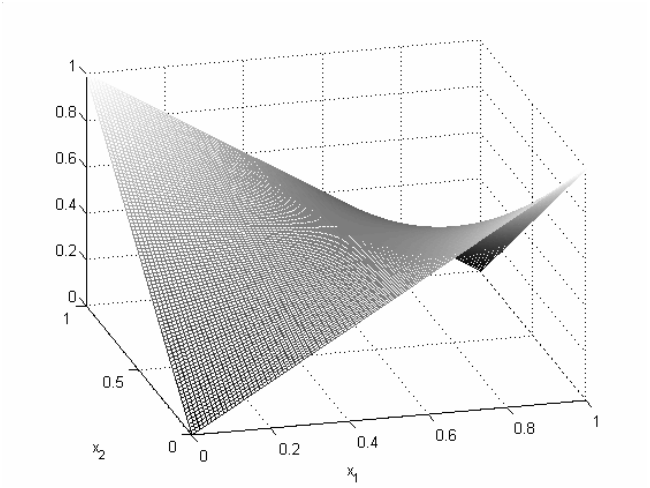


**Figure 17.** IRC output surface with three subsets per antecedent.

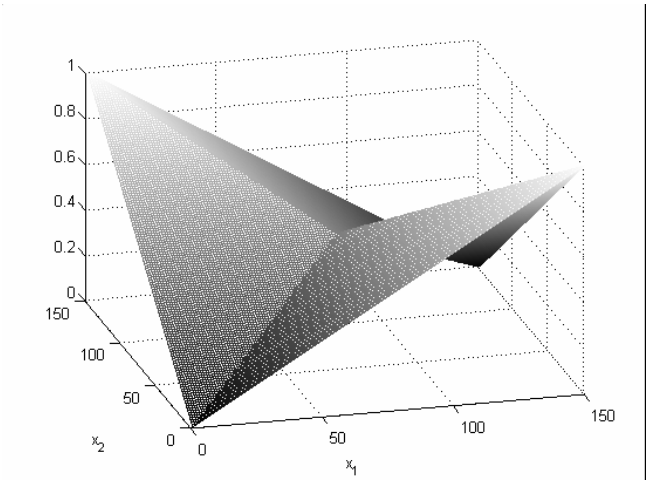


**Figure 18.** Two-layer URC system's output surface with 3 subsets per antecedent in the second layer.

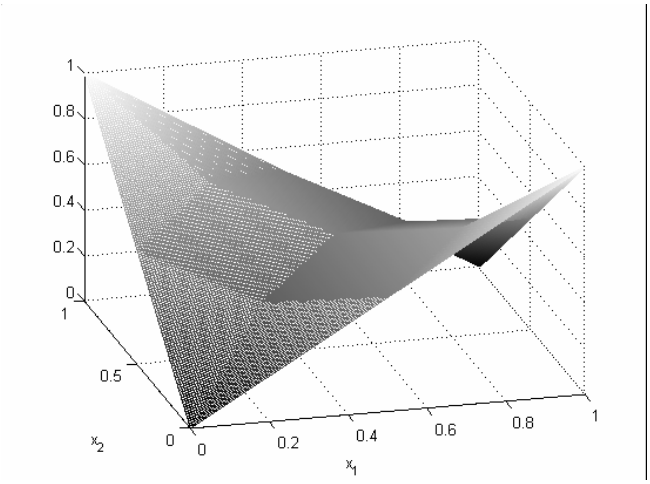


**Figure 19.** Two-layer URC system's output surface with 5 subsets per antecedent in the second layer.

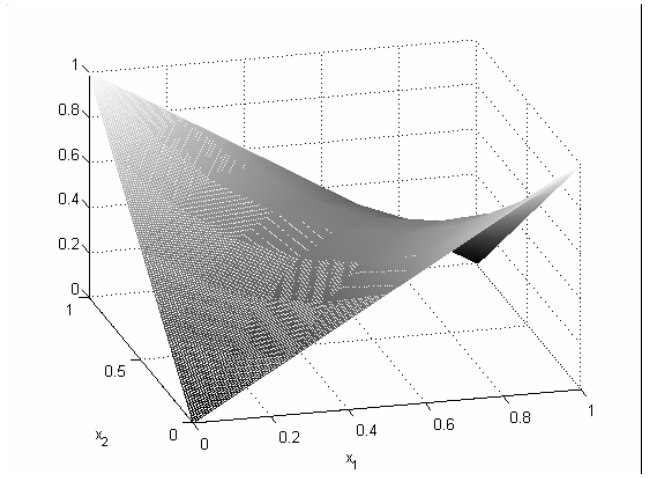


**Figure 20.** Two-layer URC system's output surface with 9 subsets per antecedent in the second layer.

The two-layer URC approximation can be improved by increasing the number of antecedent subsets. Figures 19 and 20 depict the output surface of the two-layer URC for the cases where the second layer antecedents each have 5 and 9 subsets, respectively. The number of subsets in the first URC layer remains constant. Fig. 21 shows the maximum absolute error (XAE) and the mean absolute error (MAE) for the three URC approximations. Indeed,

Indeed, the approximation error diminishes as the number of subsets increases.

| # of subsets / antecedent | 3 | 5 | 9 |
|---|---|---|---|
| XAE | 0.1250 | 0.0312 | 0.0079 |
| MAE | 0.0410 | 0.0102 | 0.0026 |

**Figure 21.** The XAE and MAE of the two-layer URC approximation for increasing numbers of subsets per antecedent in the second URC layer.

| # of Inputs \ Subsets per Antecedent | 2 | 5 | 10 | 50 |
|---|---|---|---|---|
| 2 | 4 (**8**) | 25 (**14**) | $10^2$ (**24**) | $2.5 \cdot 10^3$ (**104**) |
| 5 | 32 (**35**) | $3.13 \cdot 10^3$ (**50**) | $10^5$ (**75**) | $3.13 \cdot 10^8$ (**275**) |
| 10 | $1.02 \cdot 10^3$ (**120**) | $9.77 \cdot 10^6$ (**150**) | $10^{10}$ (**200**) | $9.77 \cdot 10^{16}$ (**600**) |
| 50 | $1.26 \cdot 10^{15}$ (**$2.6 \cdot 10^3$**) | $8.88 \cdot 10^{34}$ (**$2.75 \cdot 10^3$**) | $10^{50}$ (**$3 \cdot 10^3$**) | $8.88 \cdot 10^{84}$ (**$5 \cdot 10^3$**) |
| 100 | $1.27 \cdot 10^{30}$ (**$1.02 \cdot 10^4$**) | $7.89 \cdot 10^{69}$ (**$1.05 \cdot 10^4$**) | $10^{100}$ (**$1.1 \cdot 10^4$**) | $7.89 \cdot 10^{169}$ (**$1.5 \cdot 10^4$**) |

**Figure 22.** Comparison of the number of rules in the IRC and two-layer URC for a class of additively inseparable problems similar to the fuzzy XOR where the number of outputs in the first URC layer is equal to the number of inputs to the first layer. The total numbers of rules required by the two-layer URC inference system are shown in bold.

As IRC inference systems are crippled by rule explosion, the addition of antecedent subsets to an IRC inference system is *extremely* costly. Conversely, use of additional antecedent subsets in the two-layer URC incurs a minimal increase in complexity. A general comparison of the number of rules in the IRC and two-layer URC is impossible since the number of URC rules is dependent on the precise structure of the desired additively inseparable output surface. Fig. 10, which was originally presented as a comparison of the number of rules in an IRC and a single-layer URC, also provides a comparison of the number of rules required by the IRC system and the second layer of a two-layer URC system for problems similar in nature to the fuzzy XOR. Further, Fig. 22 provides a comparison of the IRC and two-layer URC systems under the additional assumption that the number of

of outputs from the first URC layer is equal to the number of inputs to the first layer. The layered URC significantly reduces the effect of rule explosion—indeed, a two-layer URC implementation of a MIMO problem with ten inputs that requires five subsets per antecedent in the second layer requires 99.9985% fewer rules than the analogous IRC inference system!

Finally, we observe that further complexity savings are achievable for some problems by building a multi-layer URC that contains more than two layers. In [22, 25] it was discovered that some output surfaces may require a large number of one-dimensional projections to achieve a sufficient level of reconstruction accuracy. As an example, the output surface

$$h(x_1, x_2) = \begin{cases} 1 - \frac{5}{8}x_1^2 + \frac{3}{8}x_1 x_2 - \frac{5}{8}x_2^2 & \frac{5}{8}x_1^2 - \frac{3}{8}x_1 x_2 + \frac{5}{8}x_2^2 \le 1 \\ 0 & \text{else} \end{cases} \quad (34)$$

which has form

$$h_{\pi/4}(x_1, x_2) = \begin{cases} 1 - x_1^2 - \frac{1}{4}x_2^2 & x_1^2 + \frac{1}{4}x_2^2 \le 1 \\ 0 & \text{else} \end{cases} \quad (35)$$

when rotated by $\pi/4$ radians may require a fairly large two-layer URC implementation due to the number of projections required. However a three-layer URC implementation is quite compact: the first URC layer forms two rotated variables at the angles of $\pm\pi/4$ radians; the second URC layer constructs the surface

$$z(x_1, x_2) = 1 - \frac{5}{8}x_1^2 + \frac{3}{8}x_1 x_2 - \frac{5}{8}x_2^2 \quad (36)$$

using the first layer outputs; and the third layer approximates (34) by implementing the nonlinear mapping

$$w(z) = \begin{cases} z & z \ge 0 \\ 0 & \text{else} \end{cases}. \quad (37)$$

Development of a more general theory for URC inference systems composed of three or more layers is left for future research.

## 6 Conclusions

This paper explores the relationship between the IRC and URC. A mapping is derived between the IRC and single-layer URC for additively separable problems. If the desired output surface is additively separable, rule explosion can be avoided through application of a single-layer URC. In addition, a multi-layer URC containing two or more layers is shown to be a universal approximator. Rule explosion can be avoided for some additively inseparable problems through application of a multi-layer URC inference system.

Although this paper presents simple pedagogical examples, it is not intended to be a "how-to" manual for constructing URC inference engines. However, it is important to note that a URC rule table may constructed directly from a desired output surface. It is not necessary, nor is it practical for large MIMO problems, to first construct an IRC rule table.

## References


[1] O. Kaynak, K. Jezernik, A. Szeghegyi, "Complexity reduction of rule based models: a survey," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02), Honolulu, HI, 2000, pp. 1216-1222.

[2] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," *IEEE Trans. Fuzzy Systems*, vol. 6, no. 1, pp. 1-11, Feb. 1998.

[3] J. Yi, N. Yubazaki, and K. Hirota, "A proposal of SIRMs dynamically connected fuzzy inference model for plural input fuzzy control," *Fuzzy Sets and Systems*, vol. 125, pp. 79-92, 2002.

[4] J. M. Mendel and Q. Liang, "Comments on 'Combinatorial rule explosion eliminated by a fuzzy rule configuration,'" *IEEE Trans. Fuzzy Systems,* vol. 7, no. 3, pp. 396-373, June 1999.

[5] W.E. Combs, "Author's Reply," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3., pp. 371-373, June 1999.

[6] S. Dick and A. Kandel, “Comment on ‘Combinatorial rule explosion eliminated by a fuzzy rule configuration,’” *IEEE Trans. Fuzzy Systems*, vol. 7, no. 4, pp. 475-477, Aug. 1999.

[7] W.E. Combs, "Author's Reply,” *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 4., pp. 477-478, Aug 1999.

[8] E. Trillas and C. Alsina, “On the law [pAq->r] = [(p->r)V(q->r)] in fuzzy logic,” *IEEE Trans. on Fuzzy Systems*, VOL. 10, NO. 1, pp. 84-88, Feb 2002.

[9] B.-S. Chen, H.-J. Uang, and C.-S. Tseng, “Robust Tracking Enhancement of robot systems including motor dynamics: a fuzzy-based dynamic game approach,” *IEEE Trans. Fuzzy Systems*, Vol. 6, pp 538-552, Nov 1998.

[10] H. Emara and A. L. Elshafei, “Comments on ‘Robust Tracking of robot systems including motor dynamics: a fuzzy-based dynamic game approach’”, *IEEE Trans. on Fuzzy Systems*, Vol. 10, No. 3, pp. 412-414, June 2002.

[11] B.-S. Chen, H.-J. Uang, and C.-S. Tseng, “Author’s Reply”, *IEEE Trans. on Fuzzy Systems*, Vol. 10, No. 3, pp. 414, June 2002.

[12] J. J. Weinschenk, W. E. Combs, R. J. Marks II, “Avoidance of rule explosion by mapping fuzzy systems to a disjunctive rule configuration,” *IEEE Int’l Conference on Fuzzy Systems*, St. Louis, MO, 2003, pp 43-48.

[13] J. J. Weinschenk, R. J. Marks II, W. E. Combs, “Layered URC fuzzy systems: a novel link between fuzzy systems and neural networks,” *Proc. IEEE Intl’ Joint Conf. on Neural Networks*, Portland, OR, 2003, pp. 2995-3000.

[14] L.-X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807-814, Sept. 1992.

[15] L.-X. Wang, “Fuzzy systems are universal approximators,” *Proc. IEEE Int’l. Conference on Fuzzy Systems*, San Diego, CA, 1992, pp. 1163-1170.

[16] V. Kreinovich, G. C. Mouzouris and H. T. Nguyen, “Fuzzy rule based modeling as a universal approximation tool,” *Fuzzy Systems, Modeling and Control* (H. T. Nguyen and M. Sugeno, Eds.), pp. 135-195, Kluwer Ac. Publ., Boston, 1998.

[17] M. L. Minsky, S. A. Papert, *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*. MIT Press, 1987.

[18] G. Cybenko, “Approximations by superpositions of a sigmoidal function,” *Math. Control, Signals and Systems* vol. 2, pp. 303-314, 1989.

[19] M. Stinchcombe and H. White: "Universal Approximation Using Feedforward Networks with Non-Sigmoid Hidden Layer Activation Functions," *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 612-617, 1989.

[20] D. E. Dudgeon, R. M. Mersereau, *Multidimensional Digital Signal Processing*. Prentice Hall, 1984.

[21] F. Ashrafzadeh, E. P. Nowicki, M. Mohamadian, J. C. Salmon, “Reducing the order of fuzzy logic controllers based on control surface rotation,” *Canadian Conference on Electrical and Computer Engineering*, St. John's, Newfoundland, May, 1997, pp. 785-787.

[22] J. J. Weinschenk, “Complexity reduction in fuzzy inference systems,” dissertation, University of Washington, 2005.

[23] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Pub Co., 1989.

[24] R. D. Reed, R. J. Marks II, *Neural Smithing*. The MIT Press, Cambridge MA, 1998.

[25] J. J. Weinschenk, R. J. Marks II, W. E. Combs “On the use of Fourier methods in URC system design,” *IEEE Int’l Conference on Fuzzy Systems*, Budapest, Hungary, 2004.